10.0finekit

FineReportAPIAPI

demo

IndependentChartProvider

```
package com.fr.chart.fun;

import com.fr.chart.chartattr.Chart;
import com.fr.stable.fun.Level;

/**
 * Created by eason on 15/4/21.
 *
 */
public interface IndependentChartProvider extends Level {

 String XML_TAG = "IndependentChartProvider";

 int CURRENT_API_LEVEL = 3;

 /**
 * key
 *
 * @return key
 */
 String getChartName();

 /**
 *
 *
 * @return
 */
 Chart[] getChartTypes();

 /**
 * webJS
 *
 * @return JS
 */
 String[] getRequiredJS();

 /**
 * JSdom
 *
 * @return JS
 */
 String getWrapperName();

 /**
 *
 *
 * @return
 */
 String getChartImagePath();
/**
 * API,
 *
 * @return API
 */
int currentAPILevel();

}
```

AbstractIndependentChartsProvider

DemoChartsPie extends AbstractIndependentChartsProvider

```
package com.fr.plugins.democharts.commenDataPanePie;

import com.fr.chart.chartattr.Chart;
```

```
import com.fr.chart.fun.impl.AbstractIndependentChartsProvider;


/**
 * Created by mengo on 16/5/11.
 * AbstractIndependentChartsProvider
 * getChartName
 * getWrapperName
 * getChartTypes
 * getChartImagePath
 * getRequiredJS
 * currentAPILevel
 */
public class DemoChartsPie extends AbstractIndependentChartsProvider {

    public static pieChart[] charts = new pieChart[]{
            new pieChart(),
    };

    /**
     * key
     *
     * @return key
     */
    @Override
    public String getChartName() {
        return "";
    }

    /**
     * webJS
     *
     * @return JS
     */
    @Override
    public String[] getRequiredJS() {
        return new String[]{
                "/com/fr/plugins/democharts/common/web/echarts.bridge.js"
        };
    }

    /**
     * JSdom
     *
     * @return JS
     */
    @Override
    public String getWrapperName() {
        return "EChartsFactory";
    }

    /**
     *
     *
     * @return
     */
    @Override
    public Chart[] getChartTypes() {
        return charts;
    }

    /**
     *
     *
     * @return
     */
    @Override
    public String getChartImagePath() {
        return "com/fr/plugins/democharts/customDataPanePie/images/pie256.png";
    }
```

```
    /**
     * @return API
     */
    @Override
    public int currentAPILevel() {
        return CURRENT_API_LEVEL;
    }

}
```

IndependentChartUIProvider

```
package com.fr.design.chart.fun;

/**
 * Created by eason on 14/12/29.
 *
 * @since 8.0
 *
 */
public interface IndependentChartUIProvider extends Level {

 String XML_TAG = "IndependentChartUIProvider";

 int CURRENT_API_LEVEL = 3;


/**
 * API,
 *
 * @return API
 */
int currentAPILevel();
 /**
 *
 *
 * @return
 */
 ThirdChartConfigPane getChartConfigPane(String plotID);

 ......()
}
```

AbstractIndependentChartsUI

DemoChartsPieUI extends AbstractIndependentChartsUI

```
package com.fr.plugins.democharts.commenDataPanePie;

import com.fr.design.chart.fun.impl.AbstractIndependentChartsUI;
import com.fr.design.mainframe.chart.ChartsConfigPane;


/**
 * Created by mengao on 2017/4/26.
 * AbstractIndependentChartsUI
 * getChartConfigPane
 * currentAPILevel
 */
public class DemoChartsPieUI extends AbstractIndependentChartsUI {

    /**
     * @param plotID
     * @return
     */
    @Override
    public ChartsConfigPane getChartConfigPane(String plotID) {
        return new ChartConfigPane();
    }

    /**
     * @return API
     */
    @Override
    public int currentAPILevel() {
        return CURRENT_API_LEVEL;
    }

}
```

getTableDataSourcePane(Plot plot, ChartDataPane parent)getReportDataSourcePane(Plot plot, ChartDataPane parent)chartDataDefinitionplugins-demoChart

ChartsConfigPane

```
package com.fr.design.mainframe.chart;

import com.fr.chart.chartattr.ChartCollection;
import com.fr.chart.chartattr.Charts;
import com.fr.general.Inter;
import com.fr.stable.StableUtils;

import javax.swing.*;

/**
 * Created by mengao on 2017/5/16.
 */
public abstract class ChartsConfigPane <T extends Charts> extends AbstractChartAttrPane {

    public final static String CHART_STYLE_TITLE = Inter.getLocText("Chart-Style_Name");

    public abstract Class<? extends Charts> accptType();

    @Override
    public void populate(ChartCollection collection) {
        if (StableUtils.classInstanceOf(collection.getSelectedChart().getClass(),accptType())) {
            populate(collection, (T)collection.getSelectedChart());
        }
    }

    protected abstract void populate(ChartCollection collection, T selectedChart);

    @Override
    public void update(ChartCollection collection) {
        if (StableUtils.classInstanceOf(collection.getSelectedChart().getClass(),accptType())) {
            update(collection, (T)collection.getSelectedChart());
        }
    }

    protected abstract void update(ChartCollection collection, T selectedChart);

    @Override
    protected JPanel createContentPane() {
        return new JPanel();
    }

    @Override
    public String getIconPath() {
        return "com/fr/design/images/chart/ChartStyle.png";
    }

    @Override
    public String title4PopupWindow() {
        return CHART_STYLE_TITLE;
    }
}
```

ChartConfigPane extends ChartsConfigPane

```
package com.fr.plugins.democharts.commenDataPanePie;

import com.fr.chart.chartattr.ChartCollection;
import com.fr.design.mainframe.chart.ChartsConfigPane;

import javax.swing.*;
import java.awt.*;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;

/**
 * Created by mengao on 2017/4/26.
 * ChartsConfigPane
 * populate
 * update
 */
public class ChartConfigPane extends ChartsConfigPane<PieChart> {
  private JPanel northJpane = new JPanel();
private JPanel centerJpane = new JPanel();
private JLabel nameJLable = new JLabel("");
private JButton updateButton = new JButton("");

protected JTextField value = new JTextField();
protected ChartCollection chartCollection;



public ChartConfigPane() //
{
    this.setLayout(new BorderLayout());
    northJpane.setLayout(new GridLayout(3, 2));
    northJpane.add(nameJLable);
    northJpane.add(value);
    this.add(northJpane, BorderLayout.NORTH);
    centerJpane.add(updateButton, BorderLayout.CENTER);
    this.add(centerJpane, BorderLayout.CENTER);
    this.setSize(200, 200);
    this.setVisible(true);
    updateButton.addActionListener(new ColorEventListener());
}

// EventListenerActionListener
class ColorEventListener implements ActionListener {

    @Override
    public void actionPerformed(ActionEvent e) {
        update(chartCollection, (T)chartCollection.getSelectedChart());
    }
}
@Override
public Class<? extends Charts> accptType() {
    return PieChart.class;
}

@Override
protected void populate(ChartCollection collection, PieChart selectedChart) {
    this.chartCollection=collection;
    value.setText(selectedChart.getCustomData());
}

@Override
protected void update(ChartCollection collection, PieChart selectedChart) {
    selectedChart.setCustomData(value.getText());
}
}
```

Charts

```
package com.fr.chart.chartattr;

import com.fr.base.chart.BaseChartGlyph;
import com.fr.base.chart.chartdata.ChartData;
import com.fr.chart.chartglyph.ChartGlyph;
import com.fr.json.JSONException;
import com.fr.json.JSONObject;
import com.fr.stable.web.Repository;

/**
 * Created by mengao on 2017/5/3.
 * chart
 */
public abstract class Charts <T extends ChartData> extends Chart  {
    protected T ChartData;

    public T getChartData() {
        return ChartData;
    }

    public void setChartData(T chartData) {
        ChartData = chartData;
    }

    protected abstract String getChartID();

    public Charts() {
        ChartsPlot chartsPlot = new ChartsPlot();
        chartsPlot.setPlotID(getChartID());
        this.setPlot(chartsPlot);
    }

    @Override
    public BaseChartGlyph createGlyph(ChartData chartData) {
        ChartGlyph chartGlyph = new ChartGlyph(){
            public JSONObject toJSONObject(Repository repo, double width, double height) throws JSONException {
                return toJSON(repo);
            }
        };

        setChartData((T)chartData);
        chartGlyph.setWrapperName(this.getWrapperName());
        chartGlyph.setRequiredJS(this.getRequiredJS());
        chartGlyph.setChartImagePath(this.getImagePath());

        return chartGlyph;
    }
    public abstract JSONObject toJSON (Repository repo) throws JSONException;
}
```

ChartConfig

PieChart extends Charts

```
package com.fr.plugins.democharts.commonpie;

import com.fr.chart.chartattr.Charts;
import com.fr.chart.chartdata.NormalChartData;
```

```
import com.fr.json.JSONArray;
import com.fr.json.JSONException;
import com.fr.json.JSONObject;
import com.fr.stable.web.Repository;
import com.fr.stable.xml.XMLPrintWriter;
import com.fr.stable.xml.XMLableReader;

import java.text.DecimalFormat;
import java.text.NumberFormat;


/**
 * Created by mengao on 2017/4/26.
 * chartCharts
 * getChartID
 * writeXML
 * readXML
 * toJSON
 */
public class PieChart extends Charts<NormalChartData> {
    private String customData = "demo";

    private static final NumberFormat format = new DecimalFormat("##%");


    public String getCustomData() {
        return customData;
    }

    public void setCustomData(String customData) {
        this.customData = customData;
    }

    @Override
    public void writeXML(XMLPrintWriter xmlPrintWriter) {
        super.writeXML(xmlPrintWriter);
        xmlPrintWriter.startTAG("customChartDemo")
                .attr("custom", getCustomData())
                .end();
    }

    @Override
    public void readXML(XMLableReader xmLableReader) {
        super.readXML(xmLableReader);
        if (xmLableReader.isChildNode()) {
            String tagName = xmLableReader.getTagName();
            if (tagName.equals("customChartDemo")) {
                setCustomData(xmLableReader.getAttrAsString("custom", "111"));
            }
        }
    }

    /**
     *
     * getChartData()
     */
    @Override
    //tojsongetChartData()json
    public JSONObject toJSON(Repository repo) throws JSONException {

        JSONObject jsonObject = JSONObject.create();
        jsonObject.put("series", seriesJSONArray())
                .put("title", JSONObject.create().put("text", getCustomData()));
        return jsonObject;
    }

    private JSONArray seriesJSONArray() throws JSONException {
        NormalChartData normalChartData = getChartData();
        int seriesLen = normalChartData.getSeriesCount();
        String radius = format.format(1.0 / seriesLen);
        //
```

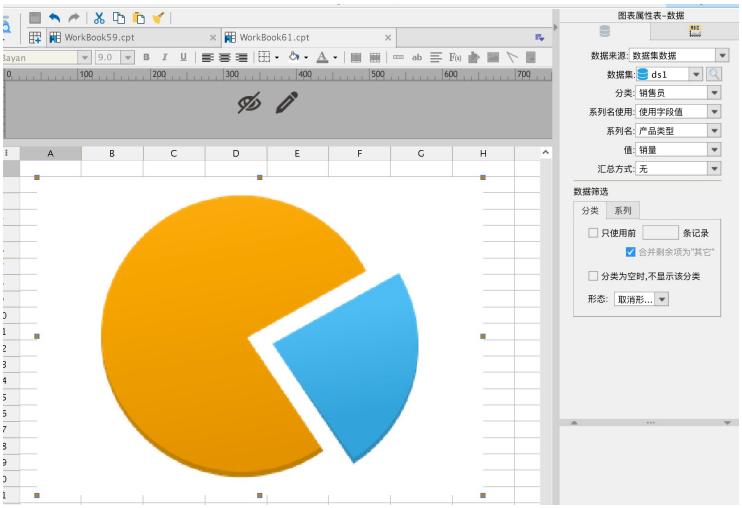
```
        JSONArray series = JSONArray.create();
        for (int index = 0; index < seriesLen; index++) {
            JSONObject wrapper = JSONObject.create()
                    .put("type", "pie")
                    .put("data", pointJSONArray(index)) //data
                    .put("name", normalChartData.getSeries_array()[index].toString());//
            if (seriesLen > 1) {
                JSONArray center = JSONArray.create()
                        .put(format.format((float) index / (seriesLen + 1) + 0.20))
                        .put("55%");
                wrapper.put("radius", radius)
                        .put("center", center);
            }
            series.put(wrapper);
        }
        return series;
    }

    private JSONArray pointJSONArray(int index) throws JSONException {
        NormalChartData normalChartData = getChartData();
        int categoriesLen = normalChartData.getCategoryLabelCount();
        JSONArray point = JSONArray.create();
        for (int i = 0; i < categoriesLen; i++) {
            JSONObject item = JSONObject.create();
            //name
            String name = normalChartData.getCategory_array()[i].toString();
            item.put("name", name);
            if (normalChartData.getSeries_value_2D().length > 0) {
                //data
                item.put("value", normalChartData.getSeries_value_2D()[index][i]);
            }
            point.put(item);
        }
        return point;
    }

    @Override
    public String getChartID() {
        return "ChartsPieWithCommenDataPane";
    }


}
```

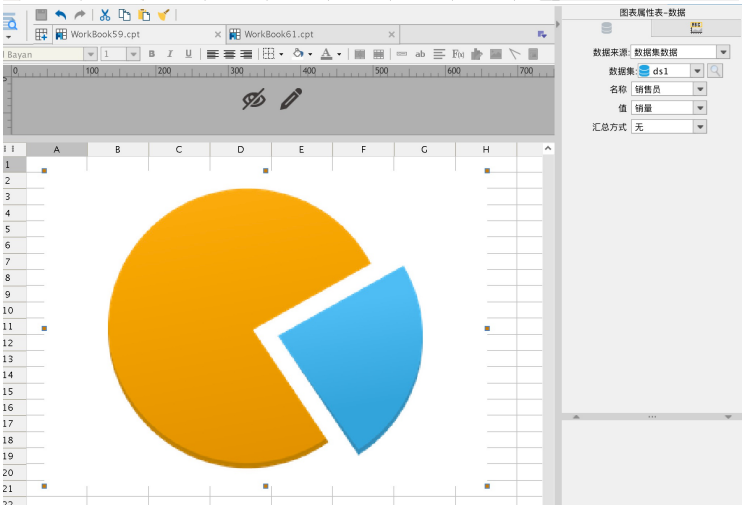
toJSONgetChartData()chartDatademo

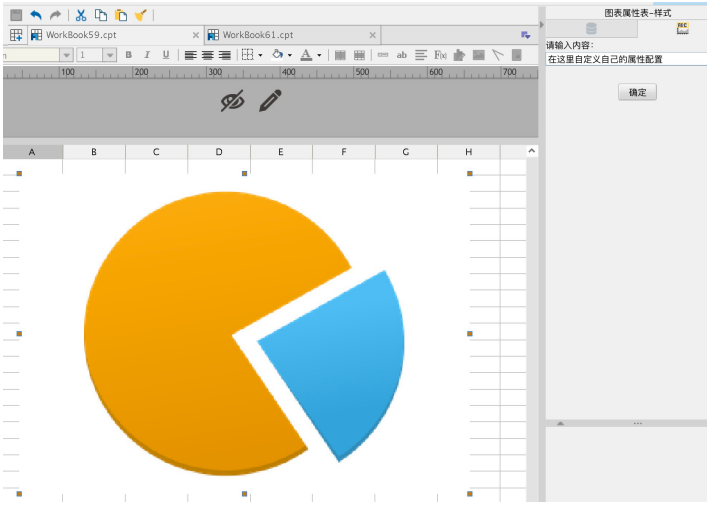
XML

plugin.xml

```
<extra-core>
    <JavaScriptFileHandler class="com.fr.plugins.democharts.common.web.EChartsFileLoader"/>
</extra-core>
<extra-chart>
 <IndependentChartProvider class="com.fr.plugins.democharts.pie.DemoChartsPie"
 plotID="DemoChartsPiePlot"/>
</extra-chart>
<extra-chart-designer>
 <IndependentChartUIProvider class="com.fr.plugins.democharts.pie.DemoChartsPieUI"
 plotID="DemoChartsPiePlot"/>
</extra-chart-designer>
```

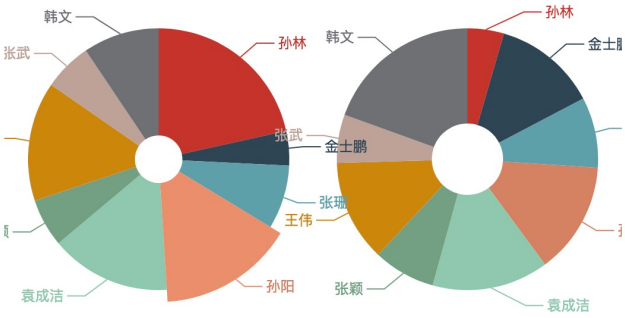


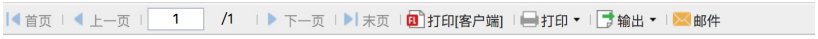




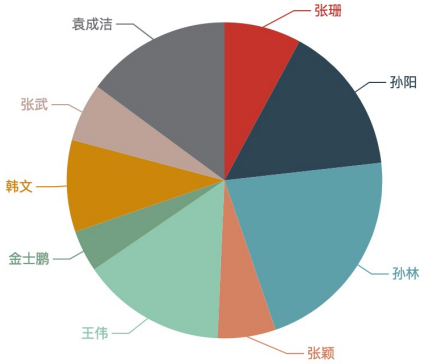
demodemodemo

公共数据配置面板demo





自定义数据配置面板demo



demo